

**INVESTR**

# **User-Interface Performance Benchmark**

**Team LAST**

May 19, 2017

## **Team Members**

Lindsay Chung

Anthony Oeum

Sean Lin (Yu Yin Lin)

## Contributors:

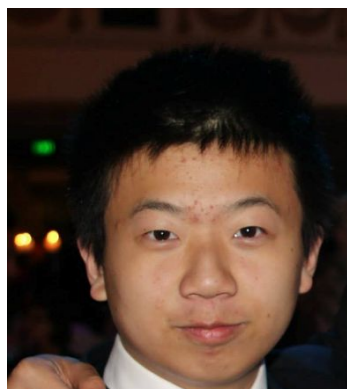
Team Member	Email	Role
Lindsay Chung	s3487579@student.rmit.edu.au	Product Owner
Anthony Oeum	s3484960@student.rmit.edu.au	SCRUM Master
Sean Lin (Yu Yin Lin)	s3486048@student.rmit.edu.au	Project Member



Lindsay Chung (Product Owner)



Anthony Oeum (SCRUM Master)



Sean Lin (Project Member)

## **Top 5 Performance Measure Factors**

For this document, Google PageSpeed Project tool was used to measure the performance issues and recommended optimisations for the Stock Market Game website project.

The first factor is to avoid landing page redirects. Apparently PageSpeed Insights detected that our website had more than one redirect from the website url to the final landing page. Best case scenario, each redirect will add a single round trip (HTTP request-response). Worst case scenario, it could result in multiple additional roundtrips to perform the DNS lookup, TCP handshake and TLS negotiation in addition to the additional HTTP request-response cycle. It is best to minimize the usage of redirects to improve the website's overall performance.

The second factor is leverage browser caching. PageSpeed Insights detected that the response from our website's server does not include any caching headers or the resources are specified to be cached only temporary. Obtaining resources over the network can be both slow and expensive; the download that is required to load the website could involve multiple roundtrips between the client and the server. This can delay processing and could block the rendering of page contents; this also incurs data costs for visitors. All the responses from the server should specify a caching policy to assist the client on determining if and when it is appropriate to reuse a previously fetched response.

The third factor is to minify CSS, HTML and JavaScript. PageSpeed Insights detected that the size of our website's resources could be reduced via minification. Minification is the process of removing unnecessary or redundant data without affecting how the resource is processed by the browser. These include code comments and formatting, removing unused code, using shorter variable and function names. To minify CSS it is suggested to use CSSNano and csso. To minify HTML, using HTMLMinifer is suggested. To minify JavaScript, it is recommended to use CloseCompiler which allows you to create a build process that uses these tools to minify and rename the development files and save them to a production directory.

The fourth factor is to enable compression. Compressing resources with gzip or deflate can reduce the number of bytes sent over to the network. Enabling compression for our website can reduce the transfer size by 61KiB (80% reduction).

The fifth factor is to reduce server response time. During the Insights test, the server responded in 0.65 seconds. Server response time measures the amount of time that it takes to load necessary HTML to begin rendering the page from the web server of the website, excluding the network latency between Google and the server. Server response times that have high variation can indicate a significant performance issue. It is recommended to reduce the server response time to be under 200ms. Some of the potential factors that causes servers to respond slowly include slow application logic, slow database queries, slow routing, frameworks, libraries, resource CPU starvation, or memory starvation. It is important to firstly measure my the server response time is so high. Gathering and inspecting existing performance and data is the first step. If there is no previous existing data on the application,

then evaluating using an automated web application monitoring solution or adding custom instrumentation can help. Secondly, identifying and fixing top performance bottlenecks is the next step. If a popular web framework or content management platform is being used then it is good to refer to the documentation for best practices for performance optimizations. Finally, monitoring and being alerted for future performance regressions is important.